Unsupervised Analysis: Dimension Reduction

Why Dimension Reduction?

For Big-Data:

- Data visualization becomes very difficult! (Cannot draw 2D scatterplots between all pairs of features).
- Big-Data often has a high degrees of redundancy. (i.e. correlation among features).
- Many features may be uninformative for the particular problem under study (noise features).
- Dimension reduction ideally allows us retain information on most important features of the data, while reducing noise and simplifying visualization & analysis.

What is Dimension Reduction?

- Map the data into a new low-dimensional space where important characteristics of the data are preserved.
- The new space often gives a (linear or non-linear) transformation of the original data.
- Visualization and analysis (clustering/prediction/...) is then performed in the new space.
- In many cases, (especially for non-linear transformations) interpretation becomes difficult.

Principal Components Analysis (PCA)

PCA

Set-up:

- Data matrix: $\mathbf{X}_{n \times p}$, n observations and p features.

Idea:

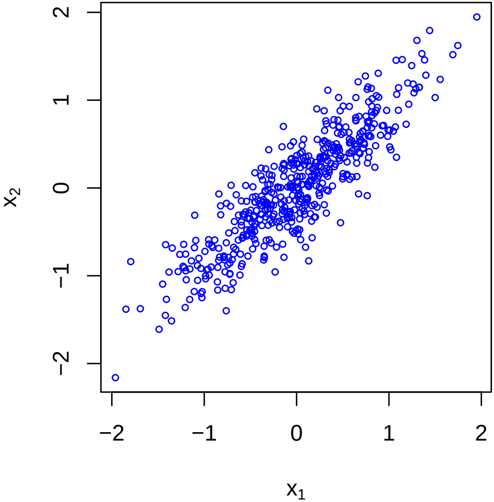
- Not all p features are needed (much redundant info).
- Find low-dimensional representations that capture most of the variation in the data.

Uses:

- Ubiquitously used - Dimension reduction, data visualization, pattern recognition, exploratory analysis, etc.
- "Best" linear dimension reduction possible.

PCA - Main Idea

Question: What is a good 1D representation of the data?



PCA - Main Idea

Some Possibilities:

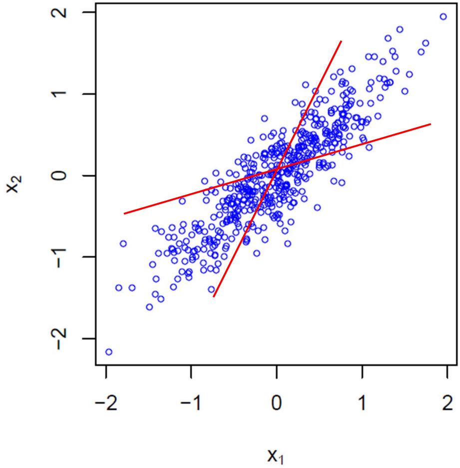
- Use one of the variables (e.g. x_1).
- Better idea: use a linear combination of the variables (i.e. a weighted average).

$$z_1 = v_1 x_1 + v_2 x_2 = \mathbf{v}^T \mathbf{X}$$

How to choose the weights (v_1 and v_2)?

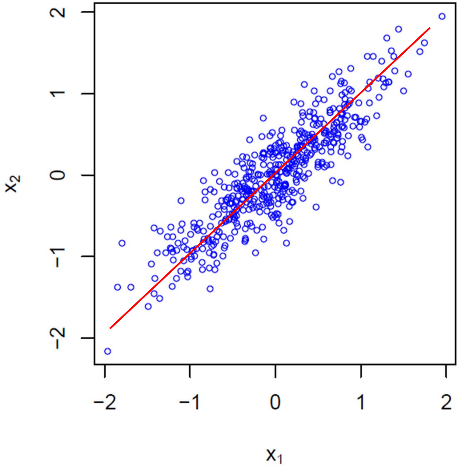
PCA - Main Idea

Many possibilities, but which one is a good choice?



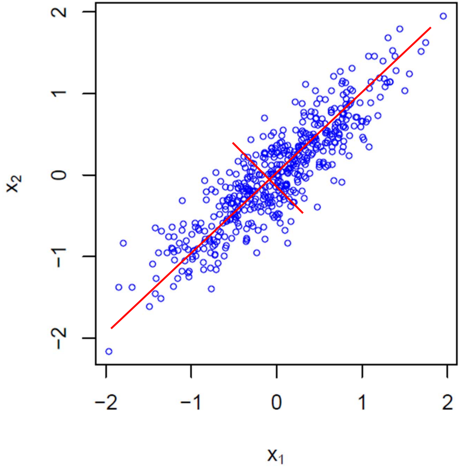
PCA - Main Idea

Find line that maximizes the variance of the data projected onto the line:

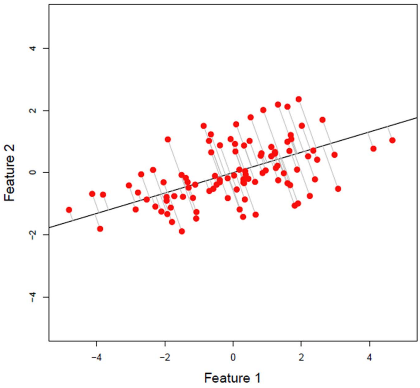


PCA - Main Idea

Subsequent components orthogonal (perpendicular).



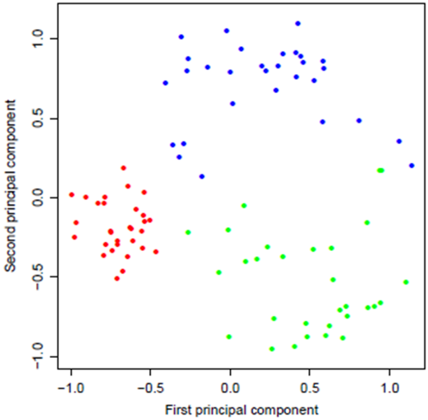
PCA - Main Idea

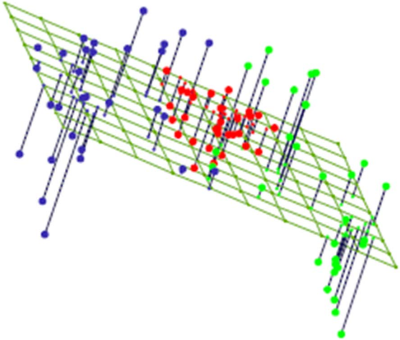


- PCA minimizes orthogonal projection onto line: $Z = v_1 x_1 + v_2 x_2$.
- Slope of line $= v_2 / v_1$ (if features centered).
- Note: Not same as OLS which minimizes projection of y onto x!

PCA - Main Idea

3D Projection onto a Hyperplane:





PCA - Criterion

PCA Criterion - PC 1 (Population):

$$\underset{\mathbf{v}}{\text{maximize}} \;\; \mathrm{Var}(\mathbf{X}\,\mathbf{v}) \;\; \text{subject to } ||\,\mathbf{v}\,||_2 = 1$$
$$\underset{\mathbf{v}}{\text{maximize}} \;\; \mathbf{v}^T\,\mathrm{Var}(\mathbf{X})\,\mathbf{v} \;\; \text{subject to } ||\,\mathbf{v}\,||_2 = 1$$
$$\underset{\mathbf{v}}{\text{maximize}} \;\; \mathbf{v}^T\,\boldsymbol{\Sigma}\,\mathbf{v} \;\; \text{subject to } ||\,\mathbf{v}\,||_2 = 1$$

where $\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{X})$.

- Finds linear combination of features that maximizes the variance.

PCA - Criterion

PCA Criterion - PC k (Population):

$$\underset{\mathbf{v}_k}{\text{maximize}} \quad \mathbf{v}_k^T \boldsymbol{\Sigma} \mathbf{v}_k \quad \text{subject to } || \mathbf{v}_k ||_2 = 1 \ \& \ \mathbf{v}_k^T \mathbf{v}_j = 0 \ \forall \ j < k.$$

- Subsequent linear combinations are orthogonal to previous combinations.
- Uncorrelated.

PCA - Criterion

PCA Criterion - Sample Version:

$$\underset{\mathbf{v}_1,\dots \mathbf{v}_K}{\text{maximize}} \quad \mathbf{v}_k^T \mathbf{X}^T \mathbf{X} \mathbf{v}_k \quad \text{subject to} \; ||\,\mathbf{v}_k\,||_2 = 1 \;\&\; \mathbf{v}_k^T \mathbf{v}_j = 0 \;\forall\, j < k.$$

Replaces $\mathbf{\Sigma}$ with estimate $\mathbf{X}^T \mathbf{X} / n$.
Solution: Eigenvalue decomposition of $\mathbf{X}^T \mathbf{X}$. (`eigen()` in `R`)

PCA - Criterion

Equivalent PCA Criterion:

$$\underset{\mathbf{u}_1,\dots \mathbf{u}_K,\mathbf{v}_1,\dots \mathbf{v}_K}{\text{maximize}} \quad \mathbf{u}_k^T \mathbf{X} \mathbf{v}_k \quad \text{subject to } ||\, \mathbf{v}_k \,||_2 = 1 \;\&\; \mathbf{v}_k^T \mathbf{v}_j = 0 \;\forall\, j < k.$$

$$||\, \mathbf{u}_k \,||_2 = 1 \;\&\; \mathbf{u}_k^T \mathbf{u}_j = 0 \;\forall\, j < k.$$

- Finds left and right projection that maximize variance.

Solution: Singular Value Decomposition (SVD) of $\mathbf{X}$. (`svd()` in R)

PCA - Parts of the Solution

SVD: $\mathbf{X}_{n\times p} = \mathbf{U}_{n\times n}\,\mathbf{D}_{n\times p}\,\mathbf{V}^T_{p\times p}$

- Singular vectors: (left) $\mathbf{U}$ and (right) $\mathbf{V}$.
  - Orthonormal - $\mathbf{U}^T\,\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\,\mathbf{V} = \mathbf{I}$.
- Singular values: Diagonals of $\mathbf{D}$.
  - $d_1 \geq d_2 \geq \ldots \geq d_r$ where $r = \mathrm{rank}(\mathbf{X})$.

SVD Solution to PCA:

- PCs: $\mathbf{Z} = \mathbf{X}\,\mathbf{V}$ or $\mathbf{Z} = \mathbf{U}\,\mathbf{D}$. ($\mathbf{U}$ are un-scaled PCs).
  - $\mathbf{z}_k = \mathbf{X}\,\mathbf{v}_k$ - k^{th} PC.
  - $\mathbf{z}_1 \ldots \mathbf{z}_K$ gives best K-dimensional projection of the data.
- PC Loadings: $\mathbf{V}$.
  - $\mathbf{v}_k$ - k^{th} PC loading (feature weights).

PCA - Properties

- Unique.
  - **U** and **V** unique up to a sign change.
  - **D** unique.
- Global Solution.

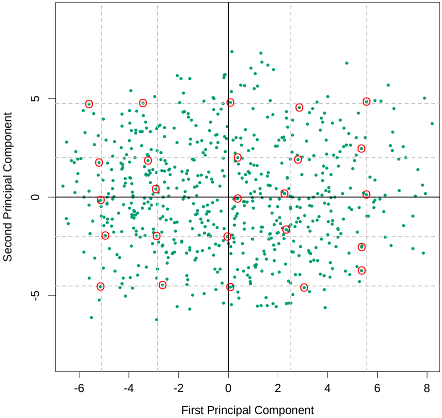
PCA - Pattern Recognition

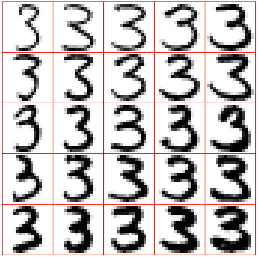
- $\mathbf{u}_1$ - first column of $\mathbf{U}$ encodes first major pattern in observation space.
- $\mathbf{v}_1$ - first column of $\mathbf{V}$ encodes the associated first pattern in feature space.
- d_1 gives strength of first pattern.

- Subsequent patterns are uncorrelated to first pattern (i.e. orthogonal).
- $\mathbf{X} \approx \sum_{k=1}^{K} d_k \, \mathbf{u}_k \, \mathbf{v}_k^T$ - data is comprised of a series of patterns.

PCA - Pattern Recognition

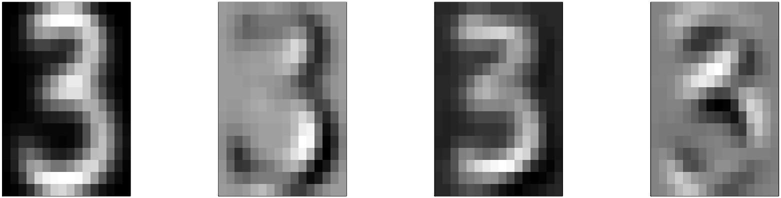
Patterns in observation space:





PCA - Pattern Recognition

Patterns in feature space:



PCA - Data Visualization

PC Scatterplots:

- Problem: Can't visualize
- Solution: Plot $\mathbf{u}_1$ vs. $\mathbf{u}_2$ and so forth.
- Advantages:
  - Dramatically reduces number of 2D scatterplots to visualize.
  - Focuses on patterns with most variance.

PC Loadings Plots:

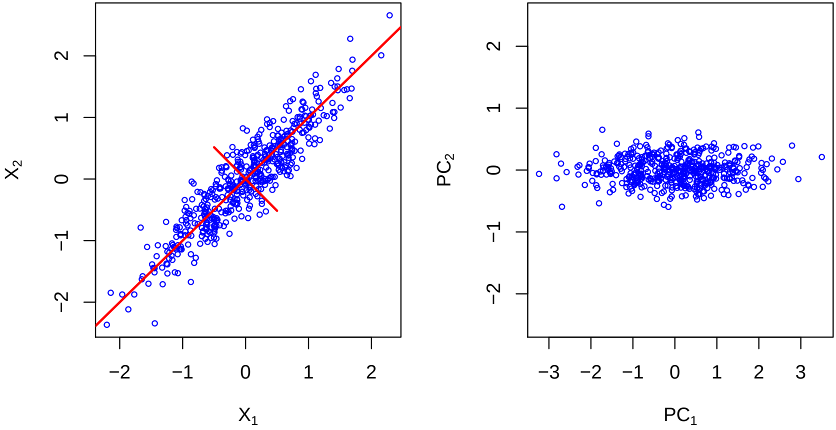
- Scatterplots of $\mathbf{v}_1$ vs. $\mathbf{v}_2$.
- Visualizations of $\mathbf{v}_k$.

Biplot:

- Scatterplot of PC 1 vs. PC 2 with loadings of $\mathbf{v}_1$ vs. $\mathbf{v}_2$ overlaid.

PCA - Data Visualization

Scatterplots:



- Plotting Scatterplot PCs roughly equivalent to rotating axes of original plot.

PCA - Dimension Reduction

Best low-rank approximation to the data:

$$\underset{\tilde{\mathbf{X}}}{\text{minimize}} \quad ||\,\mathbf{X} - \tilde{\mathbf{X}}||_F^2 \;\; \text{subject to } \text{rank}(\tilde{\mathbf{X}}) = K$$

Solution: $\tilde{\mathbf{X}} = \sum_{k=1}^{K} d_k\, \mathbf{u}_k\, \mathbf{v}_k^T$ - SVD / PCA solution!

- PCA also finds best data compression to minimize reconstruction error.
- PCA yields "best" linear dimension reduction possible!

PCA - Dimension Reduction

How much variance is explained? (i.e. extent of dimension reduction)

- Variance explained by k^{th} PC:

$$d_k^2 = \mathbf{v}_k^T \mathbf{X}^T \mathbf{X} \mathbf{v}_k \,.$$

- Total variance of data:

$$\sum_{k=1}^{p} d_k^2.$$

- Proportion of variance explained by k^{th} PC:

$$d_k^2 / \sum_{k=1}^{p} d_k^2.$$

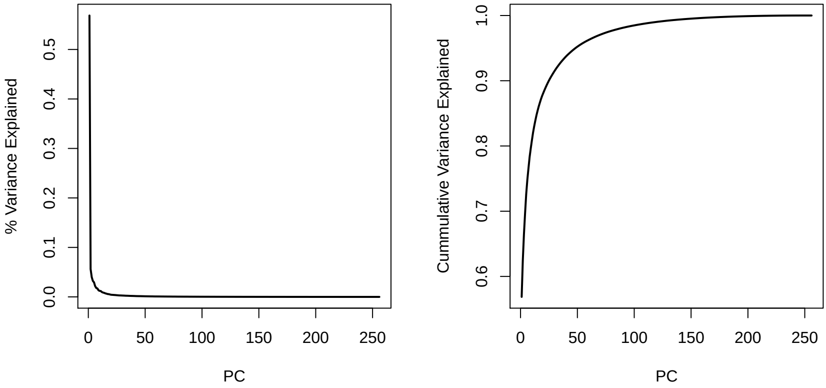
- Cumulative variance explained by first r PCs:

$$\sum_{k=1}^{r} d_k^2 / \sum_{k=1}^{p} d_k^2.$$

(Extent of dimension reduction achieve by first r PC projections.)

PCA - Dimension Reduction

Screeplot:



PCA - Dimension Reduction

How to choose K?

- Elbow in screeplot.
- Take K that explains at least 90% (95%, 99%, etc.) variance.
- More sophisticated:
  - Cross-Validation done internally.
  - Validation via matrix completion.
  - Nuclear norm penalties.

PCA - Center and Scale?

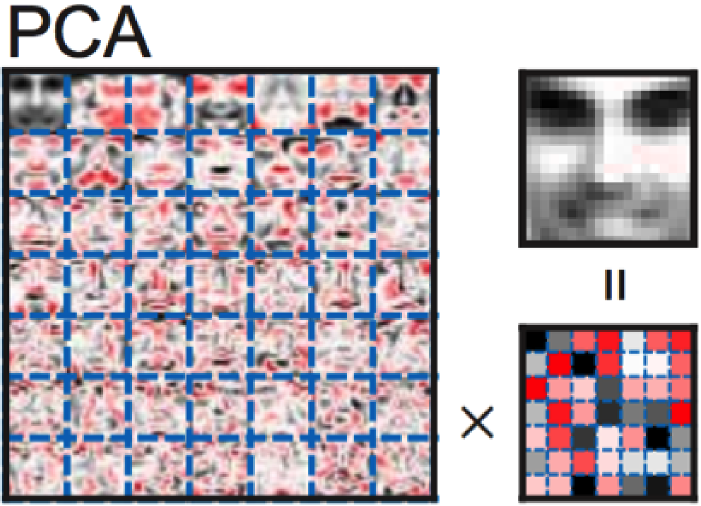
- Typically, one should center features (i.e. columns of $\mathbf{X}$).
    - Maximizing variance interpretation (assumes multivariate Gaussian model).
- Scaling changes PCA solution.
    - Features with large scale contribute more to variance, have large PC loadings.

General Suggestions:

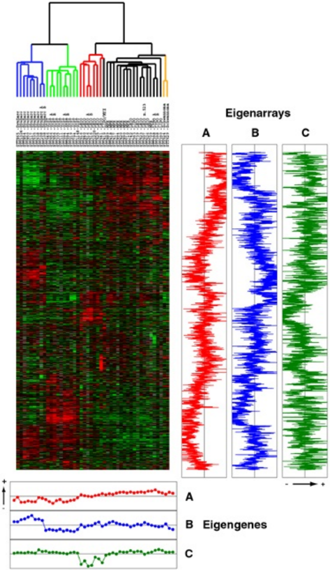
- Scale if features measured differently. (Example - US college data).
- Don't scale if features measured in same way & scale has meaning. (Example - gene expression data).

PCA - Applications

"EigenImages" or "EigenFaces"



PCA - Applications



PCA - Summary

Strengths:

- "Best" linear dimension reduction.
- Ordered / orthogonal components.
- Unique, global solution.
- Others?

Weaknesses:

- Non-linear patterns.
- Ultra-high-dimensional settings ($p >> n$)
- Others?

Sparse PCA

Motivation:

- When $p >> n$, many features irrelevant.
- PCA can perform poorly.

Idea:

- Sparsity in $\mathbf{V}$: zero out irrelevant features from PC loadings.
- Advantage: Find important features that contribute to major patterns in the data.

How?

- Typically, optimize PCA criterion with sparsity-encouraging penalty of $\mathbf{V}$.
- Many methods - active area of research!

In `R`: `SPC` in `PMA` package.

Functional PCA

Motivation:

- Times series, ordered data, spatial data.

Idea:

- Want PC loadings to be smooth (vary continuously) over time or space.
- Advantage: Improve interpretation.

How?

- Typically, optimize PCA criterion with a penalty that encourages smoothness of $\mathbf{V}$ over time or space.
- Many methods for both functional data (data in the from of curves) and discretely-sampled functional data (e.g. discrete time points or specific locations).

In R: package `fpca`.

Kernel PCA

Motivation:

- Non-linear patterns.

Idea:

- Embed inner product distances $(x_i^T x_{i'})$ in a higher-dimensional "kernel" space, $k(x_i, x_{i'})$.
- Kernel examples:
  - Radial: $k(x_i, x_{i'}) = e^{||x_i - x_{i'}||_2^2 / 2\sigma^2}$.
  - Polynomial: $k(x_i, x_{i'}) = (c x_i^T x_{i'} + 1)^d$.
- Kernel Matrix: $\mathbf{K}_{n \times n} : \mathbf{K}_{ii'} = k(x_i, x_{i'})$.
  - Idea: $\mathbf{K}$ a non-linear distance matrix.
- Find major non-linear patterns by performing PCA on $\mathbf{K}$:

$$\mathbf{K} = \mathbf{U}\,\mathbf{D}^2\,\mathbf{U}^T$$

Supervised Dimension Reduction

Partial Least Squares:

- Best dimension reduction of cross-covariance between $\mathbf{X}$ and $\mathbf{Y}$ such that factors are orthogonal to $\mathbf{X}$.

Canonical Correlations Analysis:

- Best dimension reduction of cross-covariance between $\mathbf{X}$ and $\mathbf{Y}$ such that bi-projection is orthogonal to $\mathbf{X}$ or $\mathbf{Y}$.

Linear Discriminant Analysis (classification):

- Best dimension reduction of between class covariance matrix relative to within class covariance.

Non-Negative Matrix Factorization (NMF)

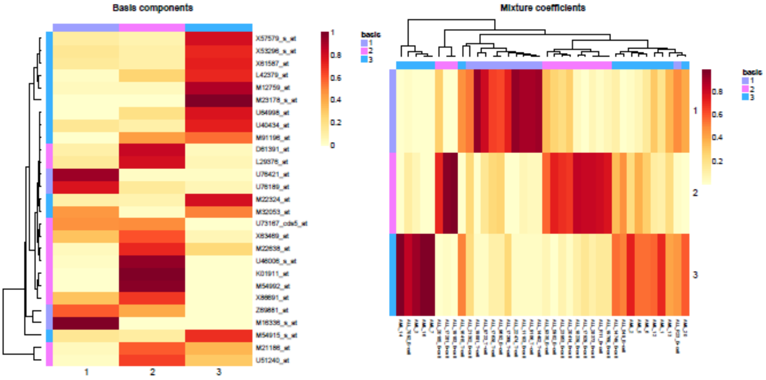
NMF

Idea: $\mathbf{X}_{n\times p} \approx \mathbf{W}_{n\times K}\,\mathbf{H}_{K\times p} = \sum_{k=1}^{K} \mathbf{W}_{:,k}\,\mathbf{H}_{k,:}$ with $K << p$.

- $\mathbf{X} \geq 0$ - non-negative data matrix.
- $\mathbf{W} \geq 0$ - non-negative observation factors; often sparse (Basis Factors).
  - $\mathbf{W}_{:,k} \geq 0$ - k^{th} observation factor.
- $\mathbf{H}_{kj} \geq 0$ - non-negative feature factors; often sparse (Mixture Factors).
  - $\mathbf{H}_{k,:} \geq 0$ - mixture of features that comprise the k^{th} factor.

Like PCA except finds patterns with same direction of correlation.

NMF



NMF Interpretation

Topic Modeling:

- $\mathbf{X}$ a matrix of news articles (rows) by words (columns) whose entries are word counts.
  - $\mathbf{X} \approx \sum_{k=1}^{K} \mathbf{W}_{:,k}\, \mathbf{H}_{k,:}$ - sum of topics.
  - $\mathbf{X}_{ij} = \mathbf{W}_{i,:}^{T}\, \mathbf{H}_{:,j}^{T} = \sum_{k=1}^{K} \mathbf{W}_{ik}\, \mathbf{H}_{kj}$.
- Topic k: Outer-product of k^{th} column of $\mathbf{W}$ ($\mathbf{W}_{:,k}$) and k^{th} row of $\mathbf{H}$ ($\mathbf{H}_{k,:}$).
  - E.g. Gay marriage.
- $\mathbf{H}_{k,:}$ non-zeros- words contributing to topic k.
  - E.g. marriage, gay, Supreme, Court, district, equal, etc.
- $\mathbf{W}_{:,k}$ non-zeros - news articles belonging to topic k.
  - E.g. "North Carolina Allows Officials to Refuse to Perform Gay Marriages" (*New York Times*).

NMF Criterion - Continuous Data

$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad ||\,\mathbf{X} - \mathbf{W}\,\mathbf{H}\,||_F^2$$

$$\text{subject to} \quad \mathbf{W}_{ik} \geq 0 \;\&\; \mathbf{H}_{kj} \geq 0$$

(PCA criterion except with non-negativity constraints.)
Algorithm Updates: (Alternating Non-negative Least Squares)

$$\hat{\mathbf{W}} = \left(\mathbf{X}\,\mathbf{H}^T(\mathbf{H}^T\,\mathbf{H})^{-1}\right)_+$$

$$\hat{\mathbf{H}} = \left((\mathbf{W}^T\,\mathbf{W})^{-1}\,\mathbf{W}^T\,\mathbf{X}\right)_+$$

Local Solution.

NMF Criterion - Count Data

$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \sum_{i=1}^{n}\sum_{j=1}^{p}[\mathbf{X}_{ij}\log(\mathbf{W}_i\,\mathbf{H}_j) - \mathbf{W}_i\,\mathbf{H}_j]$$

$$\text{subject to} \quad \mathbf{W}_{ik} \geq 0 \;\&\; \mathbf{H}_{kj} \geq 0$$

Algorithm Updates:

$$\hat{\mathbf{W}}_{ik} = \hat{\mathbf{W}}_{ik}\left(\frac{\sum_{j=1}^{p}\hat{\mathbf{H}}_{kj}\,\mathbf{X}_{ij}\,/\hat{\mathbf{W}}_i^T\,\hat{\mathbf{H}}_j}{\sum_{j=1}^{p}\hat{\mathbf{H}}_{kj}}\right)$$

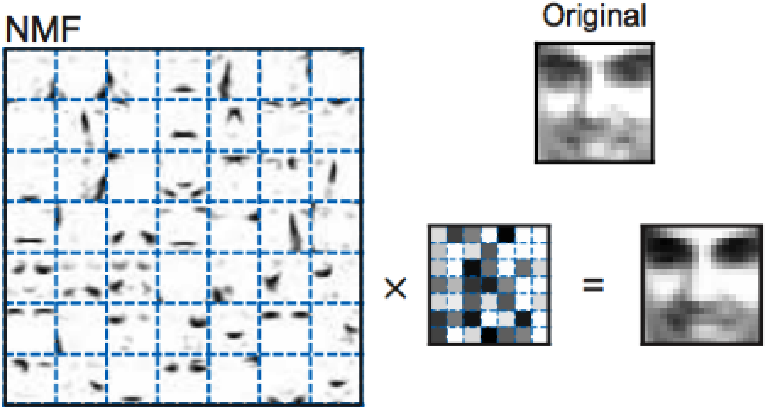
$$\hat{\mathbf{H}}_{kj} = \hat{\mathbf{H}}_{kj}\left(\frac{\sum_{i=1}^{n}\hat{\mathbf{W}}_{ik}\,\mathbf{X}_{ij}\,/\hat{\mathbf{W}}_i^T\,\hat{\mathbf{H}}_j}{\sum_{i=1}^{n}\hat{\mathbf{W}}_{ik}}\right)$$

Local solution.

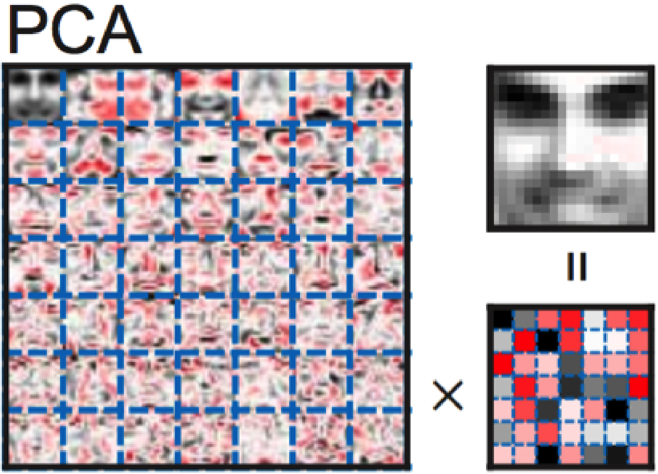
NMF - Uses

1. Dimension Reduction / Pattern Recognition.
   - Similar to PCA (e.g. component scatterplots) except that patterns of correlation found in the same direction.
2. Archetypal Analysis.
   - Caricatures (segments; contrastive categorization) vs. Prototypes (averages).
3. Soft-clustering.
   - Discussed Next Lecture!

NMF - Archetypal Analysis



NMF - Archetypal Analysis



PCA vs. NMF

Similarities:

- Linear Dimension Reduction.
- Interpretation.

Differences:

- Factors are unordered.
- Factors NOT orthogonal.
- Changing K can fundamentally change factors.
- Non-unique, non-global solution.
- Depends on initialization. (Run several times and take the best).

Choosing K

Choice depends on goal:

- Dimension Reduction:
  - Residual sums of squares (or dispersion) - Screeplot.
- Clustering:
  - Consensus, silhouette, etc. (Discussed next lecture!).
- Archetypal Analysis:
  - Sparsity, factor purity, etc.

NMF - Summary

Strengths:

- Interpretation (often more appealing than PCA!).
- Applications - Clustering & Archetypal Analysis.
- Pattern Recognition.
- Others?

Weaknesses:

- Local solutions that depend strongly on K.
- Others?

In `R`: `NMF` package.

Independent Components Analysis (ICA)

ICA

Pre-processing Step: Reduce $\mathbf{X}_{n \times p}$ to $\tilde{\mathbf{X}}_{K \times p}$ with $K < n$ # independent sources. (Typically via PCA!)

Idea: $\tilde{\mathbf{X}}_{K \times p} \approx \mathbf{A}_{K \times K} \, \mathbf{S}_{K \times p}$.

- Assumption: $\tilde{\mathbf{X}}$ a matrix of K scrambled independent signals.
- $\mathbf{A}_{K \times K}$ *Mixing Matrix* - denotes how signals are scrambled to form sources in data.
- $\mathbf{S}_{K \times p}$ *Signal Matrix* - each row of $\mathbf{S}$ is an independent signal.

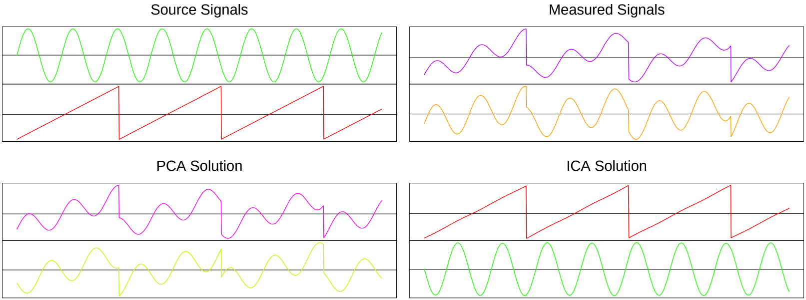
PCA finds uncorrelated, but not independent signals.

ICA Uses

1. Blind Source Separation.
   - Assume K independent signals got scrambled, but record K scrambled versions of the signal.
   - Cocktail Party Problem.
2. Denoising.
   - Noise - independent from true signals.

ICA vs. PCA

Blind Source Separation:



ICA Algorithms

Fast ICA:

- Finds rotations of $\mathbf{X}$ that are "non-Gaussian".
- Uses non-Gaussian contrast functions:
  - $g(x) = x^4$.
  - $g(x) = tanh(x)$.
- Generalization of projection pursuit.

Others:

- Infomax (entropy).

Not Statistically Independent!

PCA vs. ICA

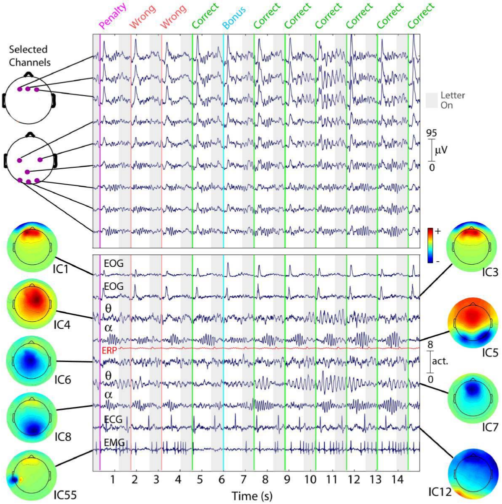
Similarities:

- Linear Dimension Reduction.
- Interpretation.

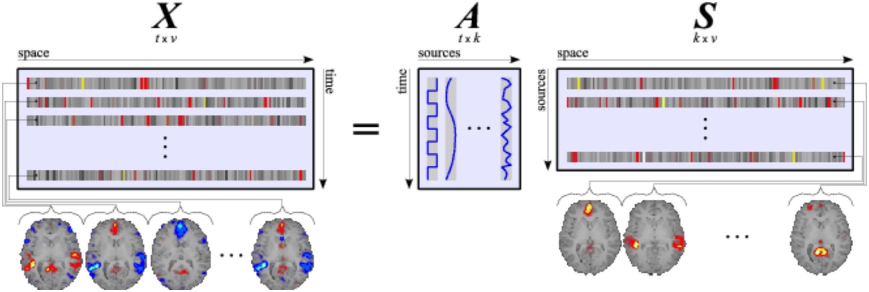
Differences:

- Factors are unordered.
- Factors NOT invariant - same solution by applying a permutation.
- Factors NOT orthogonal.
- Changing K can fundamentally change factors.
- Non-unique.
- No optimization criterion to evaluate solution.

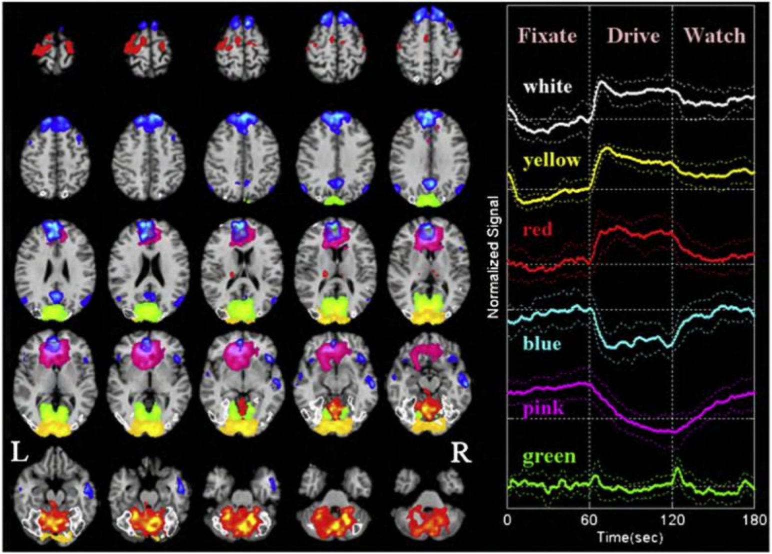
ICA Applications - EEG



ICA Applications - fMRI



ICA Applications - fMRI



ICA Summary

Strengths:

- Interpretation.
- Applications - Blind Source Separation & Denoising.
- Others?

Weaknesses:

- Solutions that depend strongly on K.
- Solutions can be rotated.
- Others?

In `R`: `fastICA` package.

Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS)

Idea:

- Visually represent proximities (similarities or distances) between objects in a lower dimensional space.
- Input: Matrix of similarities or dissimilarities, $\mathbf{D}_{n\times n}$ (don't need the data itself!).
- Goal: Find projections ($\mathbf{z}_1, \ldots \mathbf{z}_K$ where $\mathbf{z} \in \mathbb{R}^n$) that preserve original distances in $\mathbf{D}$ in a lower dimensional space ($K << n$).
- 2 Types: Classical (Metric) MDS and Non-metric MDS.
- Non-linear dimension reduction.

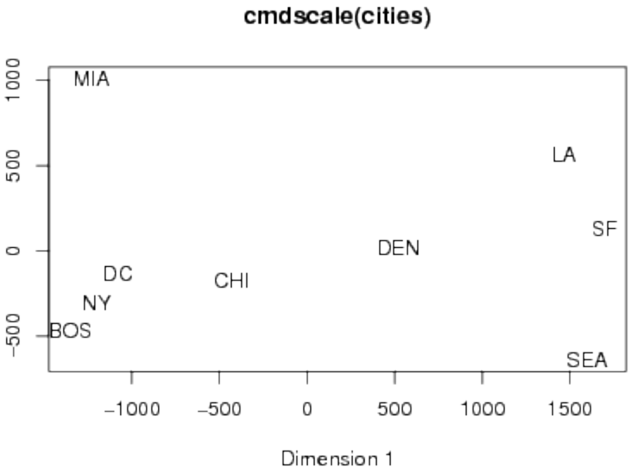
MDS - Example

Consider the distances between nine American cities:

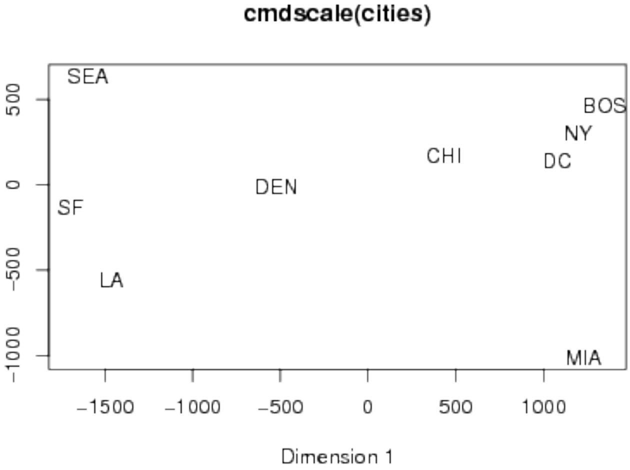
	BOS	CHI	DC	DEN	LA	MIA	NY	SEA	SF
BOS	0	963	429	1949	2979	1504	206	2976	3095
CHI	963	0	671	996	2054	1329	802	2013	2142
DC	429	671	0	1616	2631	1075	233	2684	2799
DEN	1949	996	1616	0	1059	2037	1771	1307	1235
LA	2979	2054	2631	1059	0	2687	2786	1131	379
MIA	1504	1329	1075	2037	2687	0	1308	3273	3053
NY	206	802	233	1771	2786	1308	0	2815	2934
SEA	2976	2013	2684	1307	1131	3273	2815	0	808
SF	3095	2142	2799	1235	379	3053	2934	808	0

Can we represent these cities in a 2D space like a map?

MDS - Example


cmdscale(cities)
MIA
LA
SF
DEN
DC
CHI
NY
BOS
SEA
1000
500
0
−500
−1000
−500
0
500
1000
1500
Dimension 1

MDS - Example



cmdscale(cities)
SEA
BOS
NY
CHI
DC
DEN
SF
LA
MIA
500
0
−500
−1000
−1500
−1000
−500
0
500
1000
Dimension 1

MDS - Example

Classical (Metric) MDS

- Idea: Perform PCA (eigenvalue decomposition) on the doubly centered distance matrix, $\mathbf{D}$.
  - $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix.
  - $\mathbf{z}_1, \ldots \mathbf{z}_K$ are the top K PCs of $\mathbf{H}\,\mathbf{D}\,\mathbf{H}$.
- Fact: When $\mathbf{D}$ is the matrix of Euclidean distances, classical MDS is equivalent to PCA.
  - Differences for other distance metrics.

In `R`: `cmdscale`.

Non-Metric MDS

- Idea: Optimize stress function that keeps distances in **Z** close to that of **D**.

Stress Functions:

- Least squares or Kruskal-Shephard Scaling:

$$S_D(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K) = \sqrt{\sum_{i \neq i'} (d_{ii'} - \|\, \mathbf{z}_i - \mathbf{z}_{i'} \,\|)^2}.$$

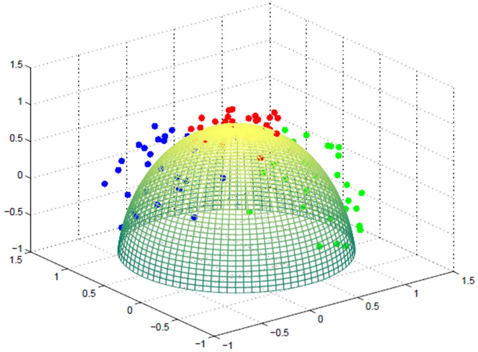
- Sammon mapping: preserve smaller pairwise distances

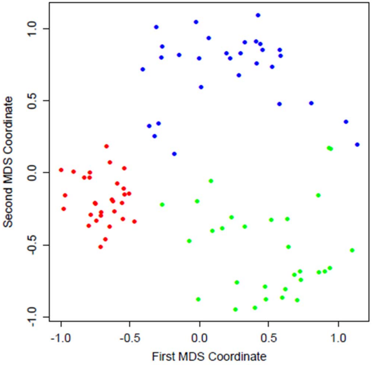
$$\sum_{i \neq i'} \frac{(d_{ii'} - \|\, \mathbf{z}_i - \mathbf{z}_{i'} \,\|^2)}{d_{ii'}}.$$

- Shepard-Kruskal nonmetric scaling ($\theta(\cdot)$: an increasing function):

$$\frac{\sum_{ii'} [\theta(\|\, \mathbf{z}_i - \mathbf{z}_{i'} \,\|) - d_{ii'}]^2}{\sum_{ii'} d_{ii'}^2}.$$

MDS - Example


1.5
1
0.5
0
−0.5
−1
1.5
1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1
1.5


Second MDS Coordinate
First MDS Coordinate
-1.0
-0.5
0.0
0.5
1.0
-1.0
-0.5
0.0
0.5
1.0

MDS Properties

- Data not needed - only dissimilarities.
- Choosing K:
  - Scree plot (like PCA).
  - Shepard Diagram - plot proximities against distances in Z.
- Interpreting MDS maps:
  - Axes and orientation arbitrary.
  - Can be rotated.
  - Only relative locations important.
  - Typically looks for objects close in the MDS map.

MDS vs. PCA

Similarities:

- Dimension reduction for visualization.

Differences:

- Non-linear vs. Linear.
- Local solution & arbitrary map.
- Non-unique & local solution.
- Only yields visualization / patterns among n objects.

MDS - Summary

Strengths:

- Visualizing proximities.
- Only need dissimilarities.
- Others?

Weaknesses:

- Arbitrary maps.
- Which stress function?
- High-dimensional settings? ($p >> n$ - more features than objects)
- Others?

In R: `dist`; `cmdscale` - classical MDS; `isoMDS` - Kruskals's MDS and `sammon` in `MASS` package.

Dimension Reduction Wrap-Up

Techniques Covered:

- PCA.
- NMF.
- ICA.
- MDS.

Dimension Reduction Wrap-Up

Comparative Strengths & Weaknesses:

Property	PCA	NMF	ICA	MDS
⋮	⋮	⋮	⋮	⋮

References

Textbooks:

- Elements of Statistical Learning by Hastie, Tibshirani & Friedman. http://statweb.stanford.edu/∼tibs/ElemStatLearn/

Some of the figures in this presentation are taken from this textbook with permission from the authors.